الفصل الثامن عشر

مفهوم برمجة الشبكات
Network Programming

Programming Concepts في هذا الفصل

18

في هذا الفصل نتناول موضوع من الموضوعات الشائكة والمتقدمة فى عالم البرمجة وهو اعداد برامج تعمل من خلال الشبكات NetWork وذلك من خلال النقاط التالية :-

- ما المقصود ببرمجة الشبكات NetWork Programming
- استعمال connections فى اعداد تطبيق خادم Server
- استعمال connections فى اعداد تطبيق مرسل Client
- استعمال DataGrams فى اعداد تطبيق خادم Server
- استعمال DataGrams فى اعداد تطبيق مرسل Client
- اعداد تطبيق تبادل الرسائل Chating

-اعداد تطبيق الخادم Server

-اعداد تطبيق مرسل Client

ما هى الشبكات؟: Networks

غرض الشبكات أساساً هو أن تتمكن جميع أجهزة الكمبيوتر من الاتصال ببعضها البعض على مسافات قريبة أو بعيدة. ما نحتاجه هو كارت شبكات وكابلات تصل بين الأجهزة واستخدام بروتوكول مثل TCP/IP.

ما المقصود ببرمجة الشبكات NetWork Programming

المقصود بذلك اعداد تطبيقات تعمل من خلال الشبكة واشهرها هو تطبيقات تبادل الرسائل Chating المشهورة وامكانية إرسال الملفات من جهاز إلى جهاز آخر.

وفى هذا الفصل نتناول فكرة اعداد تطبيقات للشبكات NetWork Programming بأكثر من طريقة وفيها نقوم باعداد نوعين من التطبيقات

الأول: تطبيق خادم Server وهو التطبيق الذى يستقبل الاتصال من الجهاز المرسل كما يستقبل منه الرسائل.

الثانى: تطبيق مرسل Client وهو التطبيق الذى يقوم بالاتصال بالجهاز الخادم Server ويرسل و يستقبل منه الرسائل.

أصبح العالم أشبة بقرية صغيرة يمكن من أي مكان في العالم أن تعرف أي أخبار تريدها أو أن تتبادل المعلومات مع أي شخص في العالم. يرجع الفضل في ذلك إلى علم الشبكات Networking والذي أمكن من خلاله وصل العالم كله. على الرغم من أن علم الشبكات Networking علم منفصل وقائم بذاته، إلا أن هناك حلقة وصل بين علم البرمجة Programming وعلم الشبكات Networking لأنه يمكنك استخدام الشبكة في بعض تطبيقات البرمجة مثل برامج الدردشة Chatting والتي تضم علمي البرمجة والشبكات في نفس الوقت.

في هذا الفصل سنبدأ إلقاء الضوء على بعض المفاهيم الخاصة بالشبكات حتى تكون معتاداً عليها وسنصل في آخر الفصل إلى بناء برنامج دردشة Chatting.

البروتوكول: Protocols

تخيل شخص أجنبي يتحدث إليك فإنك بالطبع لن تفهم منه شيئاً إلا إذا كنت تتحدث بلغته. تخيل أيضاً وجود مجموعة من الأشخاص يتحدثون بلغات مختلفة فهل يمكن التحدث معهم؟ بالطبع لا. نفس هذا الكلام بالنسبة لأجهزة الكمبيوتر فإذا لم يكن هناك مترجم ونظام يتيح للأجهزة أن تتصل ببعضها البعض وتتبادل المعلومات فإن تبادل المعلومات ببعضهم البعض يكون مستحيلاً.

ما نحتاجه هو بعض القواعد التي تتبعها الأنظمة للاتصال على شبكة. إذن البروتوكول هو القواعد المتعارف عليها بين الأنظمة المختلفة لكي تتمكن من الاتصال وتبادل المعلومات بينها.

أحد أشهر البروتوكولات هو Transmission Control Protocol Internet Protocol (TCP/IP)

وهو يتعامل مع أي نظام من أنظمة التشغيل. إذن يمكن اتصال الأجهزة بصرف النظر عن أن نظام التشغيل كان Windows, Dos, Macintosh, Unix, Linux.

ولكن تخيل وجود أكثر من جهاز كمبيوتر في الشبكة. الطبيعي أنك تريد أن تتصل بجهاز معين لتتبادل معه المعلومات. إذن كيف يمكن تحديد جهاز معين للاتصال معه في الشبكة؟ لابد أن يكون لكل جهاز عنوان خاص به لكي تتمكن من الوصول إليه. هذا العنوان يسمى IP.

عناوين الإنترنت IP addresses

كما ذكرنا يستخدم عنوان الإنترنت IP address كعنوان لجهاز الكمبيوتر في الشبكة حتى تتمكن من الوصول لجهاز معين.

يتكون عنوان الإنترنت IP address من 4 أرقام يفصل بينهم نقطة، كل رقم من الأربعة تتراوح قيمته بين صفر و255 ولذلك يسمى عنوان الإنترنت IP Address باسم Dotted quads.

مثال لعنوان الإنترنت IP address هو 130.10.32.100

لكى تعرف عنوان الإنترنت IP address لجهازك

- ☐ فى أنظمة Windows 98 و Windows NT فستجد إلى شاشة سطح المكتب أيقونة للشبكات اسمها My Network Places. انقر بالزر الأيمن بالفأرة على هذه الأيقونة واختر Properties. انقر على Protocols ثم انقر مرتين على (Internet Protocol TCP/IP) فإنك ستجد عنوان الإنترنت IP address لجهازك. بالطبع لابد أن يكون عندك كارت شبكات.
- ☐ فى أنظمة Windows 2000 فانقر باليمين على My Network Places ثم اختر Properties. ثم انقر باليمين على Local Area Connection ثم اختر Properties ثم انقر مرتين على TCP/IP Internet Protocol . ستجد عنوان الإنترنت IP address الخاص بجهازك.
- ☐ طريقة أخرى سهلة لجميع الأنظمة: افتح شاشة Dos ثم اكتب الأمر التالى ipconfig فإنك ستعرف عنوان الإنترنت IP address لجهازك.

عنوان الإنترنت IP address يحدد شيئين هما: عنوان الشبكة وعنوان الجهاز على هذه الشبكة إذن كيف نعرف كم رقم من الأربع أرقام الخاصين بعنوان الإنترنت، يحدد عنوان الشبكة وكم رقم يحدد عنوان الجهاز على الشبكة؟ نعرف ذلك عن طريق حجاب الشبكة الفرعية Subnet Mask.

حجاب الشبكة الفرعية Subnet Mask

يستخدم حجاب الشبكة الفرعية Subnet Mask لتحديد عدد الأرقام التى تحدد عنوان الشبكة من الأربع أرقام الخاصين بعنوان الإنترنت IP address وبالطبع باقى الأرقام فى عنوان الإنترنت IP address تحدد عنوان الجهاز على الشبكة. وضعت فئات لحجاب الشبكة الفرعية Subnet Mask على حسب عدد الأجهزة التى يمكن وضعها في الشبكة.

Class الفئة	IP address from عنوان الإنترنت يبدأ من	IP address to عنوان الإنترنت إلى	Subnet Mask حجاب الشبكة الفرعية	No. of Computers
A	0.0.0.0	127.255.255.255	255.0.0.0	256*256*256 ≅ 17 million
B	128.0.0.0	191.255.255.255	255.255.0.0	256*256 ≅ 65.536
C	192.0.0.0	223.255.255.255	255.255.255.0	254
D	224.0.0.0	239.255.255.255	Not Available	Not Available
E	240.0.0.0	247.255.255.255	Not Available	Not Available

مثال :

إذا كان جهازك له عنوان الإنترنت IP address (64.32.1.3) إذن فهو واقع ضمن Class A لأن الرقم الأول (64) يقع بين 0 و 128 إذن إذا نظرنا لحجاب الشبكة الفرعية Subnet Mask نجدة 255.0.0.0 إذن الرقم الأول (64) هو الذى يحدد عنوان الشبكة وبالطبع باقى الأرقام (32.1.3) تحدد عنوان الجهاز فى هذه الشبكة.

إذا كان جهازك له عنوان إنترنت IP address (130.10.32.100) إذن فهو واقع ضمن Class B لأن الرقم الأول (130) يقع بين 192,128 إذن إذا نظرنا إلى حجاب الشبكة الفرعية Subnet Mask نجده 255.255.0.0 أى أن أول رقمين (130.10) يحددان عنوان الشبكة وبالطبع (32.100) يحددان عنوان الجهاز فى هذه الشبكة.

فى Class A مثلاً نجد أن هناك 3 أرقام تحدد عنوان الجهاز وحيث أن كل رقم يقبل رقماً من صفر إلى 255 أى 256 رقم، إذن عدد الأجهزة الكلى الذى نستطيع توصيله فى شبكة من نوع Class A هو 256*256*256 أى تقريباً 17 million جهاز فى الشبكة الواحدة وبالطبع هذا الرقم كبير جداً ولذلك تستخدمه الشركات الكبيرة جداً فقط.

ونجد أن Class D,E محجوزان للتجارب المستقبلية وغير مستخدمين حالياً.

Class الفئة	Used for تستخدم فى
A	الشركات الكبيرة
B	الشركات المتوسطة
C	الشركات الصغيرة
D	غير مستخدم
E	غير مستخدم

- لا يمكن أن يتكرر عنوان الإنترنت IP address أبداً على مستوى العالم كله لأنه لو تكرر فذلك يعنى أن هناك أكثر من جهاز له نفس العنوان وذلك مستحيل.
- لا يمكن أن يكون لجهازك عنوان إنترنت IP address كلة أصفار أو كله 256 ولذلك

نجد أن عدد الأجهزة الكلي ينخفض جهازين ولذلك نجد أن Class C نستطيع توصيل 254 جهاز فقط بدلاً من 256.

- ولكن معظم الناس لا يفضلون الأرقام في التعامل مع عناوين الإنترنت فإنهم يفضلون الأسماء لأن حفظها أسهل من الأرقام ولذلك لابد من وجود بديل لعنوان الإنترنت IP address نستطيع من خلاله التعامل مع عناوين الإنترنت كنص من الحروف وليس أرقام و يتم ذلك عن طريق ما يعرف باسم (Domain Name System DNS).

DNS:

هو نظام يقوم بتحويل الاسم إلى عنوان الإنترنت IP address المناظر. إذن فإننا إذا فتحنا مستعرض الإنترنت Internet Browser مثل Internet Explorer وكتبت العنوان التالي 192.18.97.137 فإنه يكون تماماً مثل www.javasoft.com.

إذن الأسهل هو حفظ الاسم بدلاً من الأرقام إذن نظام DNS يحول الاسم إلى الـ IP address المناظر.

لاحظ أن استخدام الاثنين سواء عنوان الإنترنت بالأرقام IP Address أو بالاسم متماثل تماماً ولا فارق بينهما إلا أن استخدام الأسماء وحفظها اسهل من الأرقام كثيراً.

الآن عرفنا كيف نحدد جهاز بعينه في الشبكة ولكن المهم كيف يتم نقل البيانات؟ لابد أن نذكر اولاً ما هو الـ Port.

Port

- عادة يكون هناك كارت شبكة واحد في الكمبيوتر ولذلك يوجد توصيل واحد لكل كمبيوتر بالشبكة. جميع المعلومات والبيانات تصل إلى الكمبيوتر عن طريق هذا التوصيل الوحيد ولكن بعض المعلومات قد تكون مخصصة لتطبيقات معينة في الكمبيوتر.
- إذن كيف يستطيع الكمبيوتر معرفة التطبيق لتوصيل البيانات له؟ ذلك يتم عن طريق الـ Port.
- إذن الـ Port هو عبارة عن خدمة معينة لها رقم والبروتوكول المسمى TCP/IP يستخدم الـ Port لتوصيل البيانات الواردة إلى تطبيق معين في الكمبيوتر.

- الـ Port إذن هو رقم int يتراوح بين 0 و 65535 . ولكن من 0 إلى 1023 هى أرقام محجوزة لخدمات معينة معروفة أما الباقى فهو يستخدم للتجارب فمثلاً الـ Port رقم 21 يستخدم لنقل الملفات (FTP) والPort رقم 80 يستخدم لنقل صفحات الوب (HTTP).
- إذا أردت معرفة الخدمات Services ورقم الـ Port المناظر لها فابحث عن ملف اسمه Services بلا أى امتداد فى جهازك. افتح هذا الملف باستخدام Notepad.

أى أنك ستذهب إلى Start, Find.

Files or Folders

ثم اكتب اسم Services ستجد أكثر من نتيجة. ما يهمنا هو الملف المسمى Services بلا أى امتداد. افتح هذا الملف بالضغط مرتين عليه. سيطلب منك التطبيق الذى تريد فتح الملف به.

أختر Notepad واستعرض محتويات الملف لتعرف رقم الـ Port الخاص بكل خدمة، لاحظ أن هذا الملف لا يحتوى على جميع الخدمات ولكنه يعطيك فكرة عن الخدمات Services الموجودة.

المقابس Sockets

يتكون المقبس Socket من جزئين:

- عنوان الإنترنت IP address
- رقم الـ Port

مثلاً:إذا أردت أن تتنقل لموقع Java soft على الوب فإنك تطلب نقل صفحة الوب الخاصة بشركة Java soft اى أنك تستخدم Port رقم 80 وبالطبع لابد أن تستخدم عنوان الإنترنت IP address أو اسم الموقع Domain Name أى أنك تكتب.

http://www.javasoft.com:80

إذا لم تحدد رقم الـ Port فإن مستعرض الإنترنت يستخدم Port رقم 80 ولكنك إذا أردت استخدم Port أخر فإنك تحدد رقمة بشرط أن يكون الموقع يقدم هذه الخدمة التى تريدها.

أهم الفصائل Classes المستخدمة فى الشبكات

Class الفصيلة	وظيفته	Methods الدوال الموجودة في الفصيلة
InetAddress	نستخدمه لمعرفة عنوان الإنترنت IP address لشبكة معينة.	getLocalHost() تقوم بإرجاع هدف object من نوع الفصيلة InetAddress يمثل الجهاز الذى نستخدمه. getAddress(): تقوم بإرجاع مصفوفة Array تتكون من 4 أرقام من نوع byte تمثل عنوان الإنترنت IP address. getHostName(): تقوم بإرجاع String تمثل اسم الجهاز.
Socket	هو المستخدم من جهاز المستخدم Client لتحديد اتصال بينه والجهاز الخادم Server عن طريق تحديد عنوان الشبكة IP address والـ Port للجهاز الخادم Server.	getInputStream(): تقوم بمعرفة البيانات الداخلة إلى جهاز المستخدم Client. getOutputStream(): تقوم بمعرفة البيانات الخارجة من جهاز المستخدم Client. close(): تغلق الاتصال مع اى جهاز
ServerSocket	يستخدمه الجهاز الخادم Server ويستمع لكل مستخدم Client يحاول إجراء اتصال معه.	accept(): تنتظر إجراء اتصال بين الجهاز الخادم Server وجهاز المستخدم Client

فى لغة VB.NET

استعمال connections فى اعداد تطبيق خادم Server

الخطوة الاولى كما ذكرنا هى اعداد تطبيق الخادم Server الذى ينتظر اتصال تطبيقات العملاء Clients وينتظر منها الرسائل وفى هذه الفقرة نستعمل الطريقة الأولى وهى استعمال الاتصال المباشر Connection بينهما ويمر ذلك بخمسة خطوات هى:

1- تعريف متغير Object من الفصيلة TcpListener

كما فى السطور التالية

```
1. Dim listener As TcpListener
2. listener = New TcpListener(5000)
```

فى لغة C#

```
1. Dim listener As TcpListener
2. listener = New TcpListener(5000)
```

فى هذه السطور

- فى السطر رقم 1 تم الإعلان عن المتغير listener (منصت) من الفصيلة TcpListener
- فى السطر رقم 2 تم تخليق هدف الفصيلة TcpListener باستعمال الكلمة New
- والقيمة 500 هى رقم المنفذ Port الذى يتم الاتصال من خلاله

2- تشغيل الـ listener وذلك باستدعاء الدالة Start() كما فى السطر التالى

```
listener.Start()
```

فى لغة C#

```
listener.Start()
```

3- تعريف متغير هدف من الفصيلة Socket واستعماله فى الموافقة على عملية اتصال تطبيق العميل ويتضح ذلك من السطور التالية Clent.

```
1. Private connection As Socket
2. connection = listener.AcceptSocket()
```

فى لغة C#

```
3. Private Socket connection;
4. connection = listener.AcceptSocket()
```

- فى السطر رقم 1 تم الإعلان عن متغير هدف بالاسم connection من الفصيلة Socket.
- فى السطر رقم 2 تم استدعاء الدالة AcceptSocket() مع الهدف listener والتى تقبل اتصال تطبيق العميل والإشارة إلى الاتصال بينهما بمتغير الهدف Client Connection

4- قراءة النصوص المرسلة من الـ Client كما فى السطور التالية:

```
1. Private reader As BinaryReader
2. reader = New BinaryReader(socketStream)
3. theReply = reader.ReadString()
```

فى لغة C#

```
1. Private BinaryReader reader ;
2. reader = New BinaryReader(socketStream);
3. theReply = reader.ReadString()
```

فى هذه السطور:

- فى السطر رقم 1 تم الإعلان عن متغير هدف بالاسم reader من نوع الفصيلة BinaryReader التى توفر إمكانيات قراءة بيانات ثنائية وسوف يستعمل هذا المتغير فى قراء البيانات المرسلة.
- فى السطر رقم 2 يتم إنشاء الهدف reader مع إرسال الهدف socketStream كمعامل له لتتم القراءة منه.
- فى السطر رقم 3 يتم استدعاء الدالة ReadString() مع المتغير Reader الذى تم تعريفه فتتم القراءة.

5- الخطوة الأخيرة هى إغلاق الاتصال بين الجهاز المرسل Client والجهاز المستقبل Server كما فى السطور التالية:

```
1. reader.Close()
2. socketStream.Close()
3. connection.Close()
```

فى لغة C#

```
1. reader.Close()
2. socketStream.Close()
3. connection.Close()
```

فى هذه السطور:

- تم اخلاق متغير هدف القراءة reader ومتغير socketStream ومتغير الوصلة connection.
- وبعد عرض الخطوات المتبعة لإعداد تطبيق مستقبل Server تعال بنا نطبق معا هذه الخطوات لبناء التطبيق.

مثال

هذا المثال يوفر تطبيق مستقبل Server عند تنفيذه يعمل التطبيق وينتظر الاتصال به ولإعداده تابع الخطوات التالية:

1- قم بإعداد تطبيق Windows Application جديد.

2- قم بتصميم شكل النماذج الرئيسى Form لتصبح كما فى الشكل (1-18).

الشكل (1-18)

3- قم بتغيير اسم مربع النص الكبير الذى يظهر نتائج الاتصال وكذلك الرسائل المرسلة إلى الاسم Display وتغيير اسم مربع النص الصغير المستعمل فى كتابة الرسائل من تطبيق الـ Server إلى تطبيق Client إلى الاسم Input.

4- ادخل إلى صفحة الأوامر الخاصة بالـ Form1 وقبل تعريف فصيلة (Public Form1 اكتب السطور التالية: Class Form)

```
Imports System.Windows.Forms
Imports System.Threading
Imports System.Net.Sockets
Imports System.IO
```

فى لغة C#

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Net.Sockets;
using System.IO;
```

فى هذه السطور:

- تم استعمال الأمر imports لفتح (للإشارة إلى) المكتبات NameSpaces التى سوف نحتاجها فى البرنامج والمعرف بها الفصائل مثل System.Threading المعرف بها الفصيلة Thread وكذلك System.Net.Sockets المعرف بها فصائل التعامل مع الشبكات.

5- قم بتعريف مجموعة المتغيرات التالية داخل فصيلة Form1 كما فى السطور التالية:

```
Public Class Form1
Inherits Form
1. Private connection As Socket
2. Private readThread As Thread ' server thread
3. Private socketStream As NetworkStream
4. Private writer As BinaryWriter
5. Private reader As BinaryReader
```

فى لغة C#

```
public class Server : System.Windows.Forms.Form
 private Socket connection;
private Thread readThread;
private NetworkStream socketStream;
private BinaryWriter writer;
private BinaryReader reader
```

فى هذه السطور:

يتم تعريف مجموعة المتغيرات التى أشرنا اليها فى الخطوات السابقة.

6- قم بتعديل الدالة Sub New() لتصبح كما فى السطور التالية:

```
Public Sub New()
    MyBase.New()
    InitializeComponent()
1.  readThread = New Thread(AddressOf RunServer)
2.  readThread.Start()
End Sub ' New
```

فى لغة C#

```
public Server()
      {
         InitialiseComponent();
        readThread = new Thread( new ThreadStart( RunServer ) );
         readThread.Start();
      }
```

فى هذه السطور:

- فى السطر رقم 1 تم إنشاء الهدف readThread مع إرسال اسم الدالة التى تنفذ معه إلى دالة البناء الخاصة به لتنفذ وهى الدالة RunServer()
- فى السطر رقم 2 يتم استدعاء الدالة Start() مع الهدف readThread التى تؤدى لى بدء تنفيذ الـ Thread

7- قم بكتابة الدالة RunServer() التى تتولى عمل تطبيق الـ Server وهو الاتصال واستقبال الرسائل من تطبيق الـ Client كما فى السطور التالية:

```
Public Sub RunServer()
1.  Dim listener As TcpListener
2.  Dim counter As Integer = 1
3.  Try
4.  listener = New TcpListener(5000)
5.  listener.Start()
6.  While True
7.  Display.Text = "Waiting for connection" & vbCrLf
8.  connection = listener.AcceptSocket()
9.  socketStream = New NetworkStream(connection)
10. writer = New BinaryWriter(socketStream)
11. reader = New BinaryReader(socketStream)
```

```
12. Display.Text &= "Connection " & counter & _                " received." & vbCrLf
13. writer.Write("SERVER>>> Connection successful")
14. Input.ReadOnly = False
15. Dim theReply As String = ""
16. Try
17. Do
18. theReply = reader.ReadString()
19. Display.Text &= vbCrLf & theReply
20. Loop While (theReply <> CLIENT>>>TERMINATE" _              AndAlso
    connection.Connected)
21. Catch inputOutputException As IOException
22. MessageBox.Show("Client application closing")
23. Finally
24. Display.Text &= vbCrLf & "User terminated connection"
25. Input.ReadOnly = True
26. writer.Close()
27. reader.Close()
28. socketStream.Close()
29. connection.Close()
30. counter += 1
31. End Try
32. End While
33. Catch inputOutputException As IOException
34. MessageBox.Show("Server application closing")
35. End Try
36. End Sub ' RunServer
```

فى لغة C#

```
37. public void RunServer()
38. {
39. TcpListener listener;
40. int counter = 1;
41.
42. try
43. {
44. listener = new TcpListener( 5000 );
45. listener.Start();
46. while ( true )
47. {
48. displayTextBox.Text = "Waiting for connection\r\n";
49. connection = listener.AcceptSocket();
50. socketStream = new NetworkStream( connection );
51. writer = new BinaryWriter( socketStream );
52. reader = new BinaryReader( socketStream );
53. displayTextBox.Text += "Connection " + counter +
54. " received.\r\n";
55. writer.Write( "SERVER>>> Connection successful" );
```

```
56.
57. inputTextBox.ReadOnly = false;
58. string theReply = "";
59. do
60. {
61. try
62. {
63. theReply = reader.ReadString();
64. displayTextBox.Text += "\r\n" + theReply;
65. }
66. catch ( Exception )
67. {
68. break;
69. }
70.
71. } while ( theReply != "CLIENT>>> TERMINATE"  &&
72. connection.Connected );
73.
74. displayTextBox.Text +=
75. "\r\nUser terminated connection";
76. inputTextBox.ReadOnly = true;
77. writer.Close();
78. reader.Close();
79. socketStream.Close();
80. connection.Close();
81.
82. ++counter;
83. }
84. } // end try
85.
86. catch ( Exception error )
87. {
88. MessageBox.Show( error.ToString() );
89. }
90.
91.}
```

- فى هذه السطور يتم تعريف الدالة RunServer() التى تتولى استقبال محاولات الاتصال وتحقيق الاتصال بالـ Clients وكذلك إرسال البيانات مرة أخرى إلى الـ Client كما يلى:
- فى السطر رقم 1 تم الإعلان عن متغير من الفصيلة TcpListener كما أشرنا فى الفقرات السابقة
- فى السطر رقم 4 يتم إنشاء الهدف من الفصيلة TcpListener مع تحديد المنفذ Port رقم 5000 للاتصال.

- فى السطر رقم 5 يتم استدعاء الدالة Start() مع الهدف Listener لبداية انتظار الاتصال.
- فى السطر رقم 8 يتم استدعاء الدالة AcceptSocket() مع الهدف listener الذى تم انشاءه والإشارة إلى ذلك بالمتغير connection.
- فى السطور رقم 9و10و11 يتم إنشاء الأهداف socketStream وwriter وreader المستعملة الاتصال والقراءة والكتابة ثم يتم طباعة رسائل على الشاشة تفيد تحقق الاتصال.
- فى السطور 17،18،19و20 يوجد تكرار Loop While... DO يقوم بتكرار القراءة طالما يتحقق الاتصال ولم يكتب المستخدم الكلمة TERMINATE
- وهذا هو قلب هذه الدالة وهو الاستمرار فى القراءة حتى يتم إغلاق الاتصال

وفى باقى السطور يتم التعامل مع الأخطاء وإغلاق ما تم فتحه.

8- قم بإضافة حدث Key_Down لمربع النص Input واكتب به السطور التالية:

```
Private Sub Input_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Input.KeyDown
1.  Try
2.  If (e.KeyCode = Keys.Enter AndAlso Not connection Is Nothing) Then
3.  writer.Write("SERVER>>> " & Input.Text) '
4.  Display.Text &= vbCrLf & "SERVER>>> " & Input.Text
5.  If Input.Text = "TERMINATE" Then
6.  connection.Close()
7.  End If
8.  Input.Clear()
9.  End If
10. Catch exception As SocketException
11. Display.Text &= vbCrLf & "Error writing object"
12. End Try
13.  End Sub '
```

فى لغة C#

```
92. protected void inputTextBox_KeyDown(
93. object sender, KeyEventArgs e )
94. {
95. try
96. {
97. if ( e.KeyCode == Keys.Enter && connection != null )
98. {
99. writer.Write( "SERVER>>> " + inputTextBox.Text );
100.
101.    displayTextBox.Text +=
```

```
102.    "\r\nSERVER>>> " + inputTextBox.Text;
103.    if ( inputTextBox.Text == "TERMINATE" )
104.    connection.Close();
105.
106.    inputTextBox.Clear();
107.    }
108.    }
109.    catch ( SocketException )
110.    {
111.    displayTextBox.Text += "\nError writing object";
112.    }
113.    }
```

- في هذه السطور يتم أخذ ما يكتبه المستخدم في الـ TextBox ثم التأكد من الزر المضغوط إذا كان Enter يتم إرسال ما كتب إلى تطبيق الـ Client كما يلي.
- في السطر رقم 2 يتم اختبار ما كتبه المستخدم اذا كان الزر Enter وأيضا مازال هناك اتصال.
- في السطر رقم 3 يتم إرسال ما كتبه المستخدم في مربع النص بالإضافة إلى الكلمة Server للإشارة إلى ان مصدر البيانات هي الـ Server.
- في السطر رقم 5 يتم اختبار ما كتب إذا كان الكلمة TERMINATE يتم إغلاق الاتصال وباقي السطور كما سبق.

9- نفذ البرنامج تحصل على نتيجة التنفيذ كما في الشكل (2-18).

الشكل (2-18)

- في هذا الشكل يبدأ البرنامج عارضا رسالة تفيد الانتظار حتى يتم الاتصال.

استعمال connections فى اعداد تطبيق مرسل Client

فى هذه الفقرة نقوم باعداد تطبيق العميل Client الذى يقوم بالاتصال وإرسال الرسائل وذلك من خلال خطوات تشابه الخطوات السابقة كما يلى:

1- قم باعداد تطبيق Windows Application جديد.

2- قم بتصميم شكل النماذج الرئيسى Form لتصبح كما فى الشكل (3-18).

الشكل (3-18)

- قم بتغيير اسم مربع النص الكبير الذى يظهر نتائج الاتصال وكذلك الرسائل المرسلة إلى الاسم Display وتغيير اسم مربع النص الصغير المستعمل فى كتابة الرسائل من تطبيق الـ Server إلى تطبيق Client إلى الاسم Input.

4- أدخل إلى صفحة الأوامر الخاصة بالـ Form1 وقبل تعريف فصيلة (Public Class Form1 اكتب السطور التالية:

```
Imports System.Windows.Forms
Imports System.Threading
Imports System.Net.Sockets
Imports System.IO
```

فى لغة C#

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Threading;
using System.Net.Sockets;
using System.IO;
```

5- قم بتعريف مجموعة المتغيرات التالية داخل فصيلة Form1 كما فى السطور التالية:

```
Private output As NetworkStream
Private writer As BinaryWriter
Private reader As BinaryReader
Private message As String = "" ' message sent to server
Private readThread As Thread
```

فى لغة C#

```
        private NetworkStream output;
        private BinaryWriter writer;
        private BinaryReader reader;
        private string message = "";
```

6- قم بتعديل الدالة Sub New() لتصبح كما فى السطور التالية:

```
Public Sub New()
    MyBase.New()
    InitializeComponent()
    readThread = New Thread(AddressOf  RunClient)
    readThread.Start()
End Sub ' New
```

فى لغة C#

```
public Client()
        {
            InitializeComponent();
       readThread = new Thread( new ThreadStart( RunClient ) );
            readThread.Start();
}
```

7- قم بكتابة الدالة RunClient () التى تتولى عمل تطبيق الـ Client وهو الاتصال كما فى السطور التالية:

```
Public Sub RunClient()
1.  Dim client As TcpClient
2.  Try
3.  Display.Text &= "Attempting connection" & vbCrLf
4.  client = New TcpClient()
5.  client.Connect("localhost", 5000)
6.  output = client.GetStream()
7.  writer = New BinaryWriter(output)
8.  reader = New BinaryReader(output)
9.  Display.Text &= vbCrLf & "Got I/O streams" & vbCrLf
10. Input.ReadOnly = False
11. Try
12. Do
13. message = reader.ReadString
14. Display.Text &= vbCrLf & message
15. Loop While message <> "SERVER>>> TERMINATE"
16. Catch inputOutputException As IOException
17. MessageBox.Show("Client application closing")
18. Finally
19.  Display.Text &= vbCrLf & "Closing connection." & _vbCrLf
20. writer.Close()
21. reader.Close()
22. output.Close()
23. client.Close()
24. End Try
25. Application.Exit()
26. Catch inputOutputException As Exception
27. MessageBox.Show("Client application closing")
    End Try
End Sub ' RunClient
```

فى لغة C#

```
public void RunClient()
{
TcpClient client;
try
{
displayTextBox.Text += "Attempting connection\r\n";
client = new TcpClient();
client.Connect( "localhost", 5000 );
output = client.GetStream();
writer = new BinaryWriter( output );
reader = new BinaryReader( output );
displayTextBox.Text += "\r\nGot I/O streams\r\n";
inputTextBox.ReadOnly = false;
```

```
do
{

try
{
message = reader.ReadString();
displayTextBox.Text += "\r\n" + message;
}
catch ( Exception )
{
System.Environment.Exit(
System.Environment.ExitCode );
}
} while( message != "SERVER>>> TERMINATE" );

displayTextBox.Text += "\r\nClosing connection.\r\n";
writer.Close();
reader.Close();
output.Close();
client.Close();
Application.Exit();
}
catch ( Exception error )
{
MessageBox.Show( error.ToString() );
}

}
```

فى هذه السطور يتم تصميم الدالة الرئيسية للبرنامج والتى تنفذ داخل الـ Thread والتى تتولى الاتصال واستمرار القراءة طالما مازال يوجد اتصال ولم يكتب تطبيق الـ Server الكلمة TERMINATE كما سبق فى البرنامج السابق.

8- قم باضافة حدث Key_Down لمربع النص Input واكتب به السطور التالية:

```
Private Sub Input_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Input.KeyDown
Try
If e.KeyCode = Keys.Enter Then
writer.Write("CLIENT>>> " & Input.Text)
Display.Text &= vbCrLf & "CLIENT>>> " & _
Input.Text
Input.Clear()
End If
      Catch exception As SocketException
```

```
        Display.Text &= vbCrLf & "Error writing object"
    End Try
End Sub ' Input_KeyDown
```

فى لغة C#

```
protected void inputTextBox_KeyDown (
object sender, KeyEventArgs e )
{
try
{
if ( e.KeyCode == Keys.Enter )
{
writer.Write( "CLIENT>>> " + inputTextBox.Text );
displayTextBox.Text +=
"\r\nCLIENT>>> " + inputTextBox.Text;
inputTextBox.Clear();
}
}
catch ( SocketException )
{
displayTextBox.Text += "\nError writing object";
}

} // end method inputTextBox_KeyDown
```

ق هذه السطور:

يتم اختبار ما يكتبه المستخدم ق مربع النص Input فاذا كان الزر المضغوط هو Enter يتم استعمال الهدف Writer واستدعاء الدالة Write لإرسال ما كتب يسبقه الكلمة Client.

9- نفذ برنامج ال Server و نفذ برنامج ال Client تحصل عل نتيجة التنفيذ كما ق الشكل (4-18).

الشكل(4-18)

10- جرب كتابة رسائل بمربع نص الكتابة Input ببرنامج الـ Client ثم الضغط على الزر Enter تلاحظ إرسال الرسالة إلى تطبيق الـ Server كما فى الشكل (5-18).

الشكل (5-18)

فى لغة Java

الحزم Packages المستخدمة في الشبكات

الحزم Packages اللازمة فى الشبكات تستدعيهم عن طريق import ونستخدم هنا عدد 2 Packages للشبكات وهما:

```
import java.net.*;
import java.io.*;
```

إنشاء مقبس جهاز المستخدم Building Client Socket

لابد من إنشاء هدف object من نوع الفصيلة Socket وكما ذكرنا فإننا لابد أن نحدد عنوان الإنترنت IP address الخاص بالجهاز الخادم Server وايضاً رقم الـ Port.

مثلاً:

```
Socket clientSocket = new Socket ("130.10.32.100",8000);
```

حيث القيمة الأولى لدالة البناء constructor تحدد عنوان الإنترنت IP address الخاص بالجهاز الخادم Server وتحدد القيمة الثانية رقم الـ Port.

لاحظ استخدام رقم 8000 يمكنك بالطبع استخدام أى رقم أكبر من 1023 لأنه كما

ذكرنا فإن الأرقام من 0 إلى 1023 محجوزة ولذلك فإننا سنستخدم رقم 8000 والذي بالطبع أكبر من 1023.

القراءة والكتابة من وإلى المقبس Socket

- للكتابة إلى المقبس Socket نستخدم هدف object من نوع الفصيلة class المسماة PrintStream كالآتي:

```
PrintStream out = new PrintStream (clientSocket.getOutputStream ());
```

- لاحظ أن clientSocket هو هدف object من نوع الفصيلة class المسماة Socket كما أنشأناه في النقطة السابقة.
- أما للقراءة من المقبس Socket فإننا نستخدم هدف object من نوع الفصيلة class المسماة BufferedReader كالآتي:

```
DataInputStream in = new DataInputStream (clientSocket.getInputStream ());
```

- الدوال ()getOutputStream,()getInputStream الموجودتان في الفصيلة class المسماة Socket تستطيع استخدامهما للاتصال بين جهاز المستخدم Client والجهاز الخادم Server حيث أن الدالة ()getOutputStream تستخدم للكتابة إلى المقبس Socket والدالة ()getInputStream تقوم بالقراءة من المقبس Socket.

إغلاق أو إنهاء الاتصال

نستخدم الدالة ()close كما ذكرنا كالآتي

```
out.close ();
in.close ();
```

بناء مقبس الجهاز الخادم Building Server Socket

لابد من إنشاء هدف object من نوع الفصيلة ServerSocket وكما ذكرنا فإن بالنسبة للجهاز الخادم Server فإننا نحدد رقم الـ Port فقط الذي يجب أن يكون متفقاً تماماً مع رقم الـ Port لجهاز المستخدم Client.

السؤال الآن: لماذا نحدد عنوان الإنترنت IP address للجهاز الخادم Server ولا نحدد عنوان الإنترنت IP address لجهاز المستخدم Client.

الإجابة: تخيل أنك مدرس ولديك طلبة. الطلبة يسألونك لأنك مدرس فالكل يعرفك ويعرف اسمك ويستطيع أن يجدك في المدرسة. ولكنك لا تعرف أسماء كل الطلبة عندك ولكن من يسألك سوف تجيبه أيضاً تستطيع القول بأن نظام المستخدم/ الخادم Client/Server يماثل هذا الموقف فالجهاز الخادم Server هو المدرس والطلبة هم المستخدم Client والمدرسة هي الـ Port فلابد أن يكون الـ Port واحد بين المستخدم Client والخادم Server كما لابد أن يلتقي المدرس والطلبة في نفس المدرسة فلا يمكن أن ينتظر المدرس الطلبة في مدرسة ويذهب الطلبة إلى مدرسة أخرى، وأيضاً الجهاز الخادم Server يستقبل طلبات المستخدم Client ويجيب عنها إذن أجهزة المستخدم Client لابد أن تعرف عنوان الجهاز الخادم Server ولكن الجهاز الخادم Server لا يهمه عنوان المستخدم Client فكل مستخدم Client يطلب منه خدمة فإنه يعطيه كالمدرس بالضبط.

ننشئ هدف object من نوع ServerSocket كالآتي:

```
ServerSocket server = new Socket Server (8000);
```

لاحظ استخدام نفس رقم الـ Port

الاستثناءات Exceptions الممكن حدوثها في الشبكات

من الممكن كتابة عنوان إنترنت IP address خاطئ للجهاز الخادم Server وهذا يسبب استثناء Exception.

الاستثناءات Exceptions لجهاز المستخدم Client هما UnknownHostException إذا كان عنوان الإنترنت IP address للجهاز الخادم Server غير صحيح والاستثناء IOException إذا حدث خطأ في القراءة أو الكتابة من أو إلى المقبس Socket.

الاستثناء Exception للجهاز الخادم Server هو IOException إذا حدث خطأ في القراءة أو الكتابة من أو إلى المقبس Socket.

الاستماع إلى أجهزة المستخدمين Clients

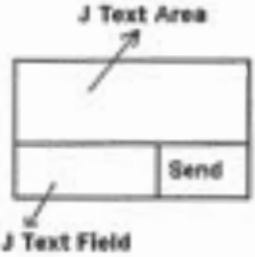
كما ذكرنا من قبل فإننا تستخدم الدالة المسماة ()accept الموجودة في الفصيلة class المسماة ServerSocket كالآتي.

```
Socket client = serverSocket.accept ();
Connection con = new Connection (client);
```

الآن نريد أن ننشئ تطبيقاً لبيان قوة لغة الجافا Java في الشبكات Networks. التطبيق هو برنامج دردشة Chatting بسيط فهيا إلى شرح المثال.

مثال 1: بناء برنامج دردشة Chatting

- نريد أن ننشئ تطبيقاً كما بالشكل.



- نفس هذا التطبيق سيتكرر مرة أخرى بنفس الشكل في الجزء القادم.
- التطبيق الأول هو الخاص بالجهاز الخادم (Server) هذا التطبيق لابد أن يكون قادراً على استقبال رسائل من أي شخص في الشبكة وكتابتها في أداة النص متعددة الأسطر JTextArea وفي نفس الوقت يستطيع إرسال الرسالة المكتوبة في أداة النص JTextField إلى الشخص الذي يتحاور معه.

إذن ما نريد ان ننشئه هو برنامج دردشة بسيط بحيث أنك تكتب الرسالة التي تريد إرسالها في أداة النص JTextField وعند الضغط على زر Send فإنك ترسل الرسالة. أي رسالة تستقبلها أو ترسلها يتم طباعتها في أداة النص متعددة الأسطر JTextArea.

نفس هذا الكلام يتم تطبيقه على الشخص الذي تتحاور معه ولكننا سننشيء برنامجاً على طريقة Client/Server بمعنى أن الجهاز الخادم سيكون قادراً على استقبال الرسالة من أكثر من شخص وجهاز المستخدم سيرسل فقط الرسالة ويستقبلها من الجهاز الخادم.